Human-to-Computer Interfaces

The present invention relates to human-to-computer interfaces and particularly, but not exclusively, to an interface system and data input apparatus, both for facilitating a reduction in the number of physical keying events required to create or delete a given data string (i.e. mnemonics, abbreviations, words, sentences, paragraphs etc.) and for providing improved calculator functionality.

The industry standard keyboard layout that possesses a virtually complete monopoly is the QWERTY keyboard. The QWERTY keyboard is a throwback to the days of mechanical typewriters and was designed to maximise the separation of the most frequently used key combinations in order to reduce jamming of the typewriter mechanism. Consequently, the keys that are most frequently used in combination are not arranged with ease of accessibility in mind and productivity is adversely affected.

Alternative keyboard models to the QWERTY layout are available, e.g. the DVORAK and MALTRON® models. These alternatives have sought to overcome the problems associated with QWERTY by respectively re-positioning the most frequently used letters of the English language in the "home row" and by curving the keyboard to fit natural finger movements. Whilst these alternative models have succeeded in increasing typing speed and reducing muscle related fatigue and stress, they have remained in the minority due to the difficulties associated with users relearning or adjusting to an unfamiliar keyboard orientation. Consequently, keyboards have continued to develop predominantly around the familiar QWERTY layout.

No alternative keyboard targeted for either the main (mass) market or specific (niche) markets actually reduces the amount of typing, and to this day, still require the same amount of typing as does a conventional QWERTY keyboard.

The growing demand for rapid data entry into computers and the increase in complex combinations of keystrokes required by modern software applications have been the driving factors behind the development of ergonomic keyboards that maximise user comfort. Several attempts have been made to achieve this goal for both able and disabled users through the appropriate positioning of keys, manipulation of keyboard consoles (i.e. splitting the console into left-hand and right-hand portions)

and the implementation of ergonomic contours for comfortable hand and finger placement.

Such improvements to keyboard design have succeeded to a limited extent in improving user comfort but to date have failed to couple this with significant improvements to keystroke efficiency and flexibility. Comfort is a palliative benefit. The only effective way to improve ergonomics and prevent injury is to do less of any activity, e.g. to reduce the amount of keyboard typing.

Computer keyboard drivers are essential in all operating system (OS) environments, their function being to convert keystrokes to OS language tables, thus bridging or translating required notation within all human-to-computer interfaces. It is important to note that the keyboard driver is a critical element to keyboard function and operation.

Conventional keyboard drivers merely map key legends to OS language tables with little or nothing in the way of sophisticated extensions or add-ons to improve performance, versatility and adaptability of the keyboard medium.

According to a first aspect of the present invention there is provided an interface system for a personal computer comprising an array of data input keys having multi-character indicia, said interface system further comprising: data storage means; data processing means; and data display means, wherein

the data processing means is adapted to facilitate a reduction in the number of key presses required to create a given data string to less than the number of characters within said data string by:

(i) filtering data stored within the data storage means by initial character, as determined by the character or characters ascribed to a data input key initially pressed by a user;

(ii) prioritising said filtered data in real-time according to user-configurable prioritisation parameters; and

(iii) displaying one or more prioritised data strings on the data display means for subsequent selection by the user.

Preferably, successive key presses act to filter further the number of data strings displayed on the data display means for subsequent selection by the user.

Preferably, the data input keys within the array have multi-character indicia which are selected to accord with a statistical extrapolation of the most used alphanumerical character combinations in a given language, to thus facilitate a further reduction in the number of key presses required to create a given data string.

Preferably, the data input keys having multi-character indicia are composite keys having at least

primary and secondary indicia corresponding to primary and secondary key-values or key-functions.

Preferably, the data storage means is defined by one or more data dictionaries in which qualitative and/or quantitative information is stored in relation to each data string.

Preferably, a configuration means is provided to allow a user to selectively enable or disable physical interactivity reduction characteristics of the interface system which facilitate a further reduction in the number of key presses required to create a given data string.

Preferably, the physical interactivity reduction characteristics are selectable from the group comprising:

(i) entering a space after selection of a data string;

(ii) limitation of displayed data strings to those having a total number of characters greater than the number of key presses required to display said data string on the data display means;

(iii) expanding typed or selected mnemonics, abbreviations or acronyms into their corresponding full data strings;

(iv) performing two-way translations between data strings and user-configurable dictionary definitions or descriptions.

(v) enabling the selection of a secondary key-value or key-function by means of double-pressing a data input key;

(vi) enabling the selection of a different data string stored within the data storage means upon each of a multiple number of presses of a data input key up to n times, said data string having an initial letter or letters corresponding to the key-value of that key; and

(vii) enabling the right-to-left and/or left-to-right deletion of n characters, words, sentences or paragraphs by means of a single key press.

Preferably, the secondary key-value or key-function obtained by double pressing a data input key is identical with the SHIFT value of that key.

Preferably, each double-press must be completed within a predetermined period of time in order to select the secondary key-value or key-function.

Preferably, the secondary key-value corresponds to the secondary indicia of a composite key having multi-character indicia.

Alternatively, the secondary key-value corresponds to a capitalised conventional key-value.

Alternatively, there is provided at least one function key operable in combination with a

composite key and adapted to access the secondary key-value or key-function.

Preferably, once successive multiple presses of a data input key cycle through all data strings retrieved from the data storage means, a further press loops back to the first data string.

Preferably, the ability to select a different data string stored within the data storage means by means of multiple presses of a data input key is disabled upon pressing of the SPACE key, or another non-character key.

Optionally, the length of the data string selectable by each successive multiple press is at least n+1 characters in length.

Preferably, the data strings selectable by each successive multiple press are actively prioritised within the data storage means according to frequency of inputting or selection.

Preferably, the multiple-press functionality overrides the double-press functionality if both are enabled simultaneously by a user.

Preferably, the configuration means also allows a user to selectively adjust the prioritisation parameters according to the desired qualitative and/or quantitative characteristics of the data stored within the, or each, data dictionary.

Preferably, the qualitative and/or quantitative information comprises statistical and/or probability information relating to each data string stored within the data storage means.

Preferably, all qualitative and quantitative information is dynamically updated in real-time.

Optionally, the data processing means maintains lookup chains between two or more data dictionaries such that a given data string in a first data dictionary is mapped to a data string or strings in one or more other data dictionaries for selection by the user.

Preferably, where a given data string in a first data dictionary is mapped to a plurality of data strings in one or more other data dictionaries, said data strings are prioritised via the configuration means for ease of selection by the user.

Preferably, the mapping is performed dynamically.

Optionally, the data processing means maintains associative links between any given data string and up to n other data strings to thus display or project the most relevant longer data string comprised of n+1 data strings for selection by the user.

Optionally, a plurality of the most relevant longer data strings are made available or displayed in a prioritised list for selection by the user.

Optionally, selection of a longer data string induces a repetition of associative linking such that a further one or more relevant longer data strings are displayed for selection by the user.

Preferably, the relevance/prioritisation of the, or each, longer data string is determined according to statistical and/or probability information stored within the, or each, data dictionary.

Preferably, statistical information relates to the historical inputting and/or selection of data strings.

Preferably, the historical inputting and/or selection information can be one or more of the following: (i) frequency of inputting; (ii) frequency of selection (iii) character length; (iv) lexical pattern density; and (v) chronological weighting.

Preferably, probability information can be one or more of the following: (i) occurrence and/or association ratios of two or more data strings within a longer data string; (ii) context ratios determining the likelihood of a given data string being grouped with one or more other data strings to determine the context of a longer data string.

Optionally, the data processing means can selectively bypass or reset the dynamically updated qualitative and quantitative information.

Preferably, the one or more data strings displayed on the data display means for subsequent selection by the user are displayed in list format in descending order of priority.

Preferably, synchronisation of data dictionaries between two or more personal computers can be accomplished by means of wired or wireless connectivity.

Alternatively or additionally, synchronisation of data dictionaries between two or more personal computers can be accomplished by means of downloading from a common database.

Preferably, the, or each, data dictionary is manually populated.

Alternatively, the population of each data dictionary with data and its corresponding qualitative and/or quantitative information may be accelerated by uploading onto the data storage means data strings resident on a personal computer or a remotely connected device.

Alternatively, the dictionaries are populated by optically scanning external data strings by means of scanning apparatus.

According to a second aspect of the present invention there is provided data input apparatus for a personal computer comprising an array of data input keys having multi-character indicia, said apparatus adapted to facilitate a reduction in the number of key presses required to create or delete a given data string to less than the number of characters within said data string.

Preferably, the multi-character indicia comprise a combination of alphabetic characters.

Preferably, the multi-character indicia include digraphs.

Alternatively or additionally, the multi-character indicia include tri-graphs.

Alternatively or additionally, the multi-character indicia include tetra-graphs.

Preferably, the keys within the array are arranged such that the most frequently used multi-character combinations in a given language are positioned closest to the home keys.

Preferably, the keys having multi-character indicia are composite keys having at least primary and secondary indicia.

Preferably, the keys having multi-character indicia are provided substantially centrally on a QWERTY keyboard having home keys F and J, respectively.

Alternatively, the keys having multi-character indicia are provided on a DVORAK or MALTRON® keyboard.

Optionally, the array of keys are represented on a graphical touch screen.

Preferably, the multi-character indicia on the graphical touch screen are dynamically updated in real time such that the multi-character combinations keyed most frequently by a user are positioned closest to the home keys.

According to a third aspect of the present invention there is provided data input apparatus for a personal computer having calculator functionality, said apparatus comprising an array of conventional numerical and calculator operator keys, a plurality of calculator control-keys and display means located on the input apparatus, wherein said control-keys are operable in combination with said calculator operator keys and/or said numerical keys to: (i) selectively send calculator-related key values to a computer; and (ii) selectively perform mathematical

calculations and display the results of said calculations on the display means and/or send said results to a computer.

Preferably, the calculator operator key values are selectable from the group comprising ., +, -, /, *, %, √, +/-, C/AC, MKUP, SEND and ENTER.

Preferably, the calculator control-keys can toggle between activated and deactivated states.

Preferably, the calculator control-keys comprise: (i) a first control key for selectively displaying the results of calculations performed using the array of numerical and calculator operator keys on the display means; and (ii) a second control key for selectively sending the results of calculations performed using the array of numerical and calculator operator keys to a computer.

Preferably, the second control key is the SEND key which, when pressed, acts to send the value displayed on the display means to the computer.

Preferably, by pressing the ENTER key, the calculator performs the most recent calculation and updates the display means accordingly without sending same to the computer.

Preferably, when both the first and second control keys are in deactivated states the conventional numerical and/or calculator operator key values

themselves are sent to a computer without performing mathematical calculations.

Preferably, the apparatus is provided with a retention buffer, which holds a calculation history of n most recent numeric entries, operators and equated values.

Preferably, the retention buffer allows a user to regress, recur and/or rectify calculations from any previous point within the buffer history.

According to a fourth aspect of the present invention there is provided data input apparatus for a personal computer comprising an array of data input keys, said apparatus adapted to facilitate a reduction in the number of key presses required to create a given data string to less than the number of characters within said data string; and wherein the apparatus comprises one or more function-lock keys that are selectable by a user to lock the functionality of the data input keys in one of two modes to maintain said selected mode until a subsequent de-selection of said function-lock key by the user.

Preferably, the function lock keys are chosen from the group comprising: ALT Lock, CTRL Lock, SEQ Lock and DUAL Lock.

Preferably, the SEQ Lock key allows the selection of secondary key-values by means of sequential as opposed to simultaneous key presses.

According to a fifth aspect of the present invention there is provided an interface system for a personal computer comprising data input apparatus according to the second aspect.

Embodiments of the present invention will now be described, by way of example only, with reference to the following drawings in which:

Fig. 1 is a perspective view of a conventional computer keyboard;

Figs. 2a and 2b are plan views of an example keyboard according to the second and third aspects of the present invention;

Fig. 2c shows the substantially centrally located keys having multi-character indicia of Figs. 2a and 2b in isolation;

Fig. 2d is a table listing the physical features of the keyboards of Figs. 2a and 2b;

Fig. 3 is a list of Internet regulated top-level domain (TLD) country codes;

Fig. 4 shows three examples of internet URL addresses typed using the composite keys shown in Fig. 2c;

Figs. 5a-i are graphs showing statistical information relating to the most frequently occurring words and lexical fragments in the English language;

Figs. 6a-e are tables depicting first and second composite key configurations respectively of the keyboards shown in Figs. 2a and 2b;

Fig. 7 is a plan view of a calculator portion of the keyboards shown in Figs. 2a and 2b according to the third aspect of the present invention;

Fig. 8 is a table showing examples of statistical extrapolations of the most commonly occurring language components for the English, French, German, Italian and Spanish languages;

Fig. 9 is a table showing examples of the manual operations and overrides for multi-press mode, translation mode and forward and backward translations where the latter two translator modes are implemented with given or highlighted text;

Fig. 10 shows two tables illustrating the mapping of key press events in a FIFO buffer;

Fig. 11 is a table illustrating the key-value mappings between various keyboard layouts;

Figs. 12a-d are plan views of alternative embodiments of the invention having alternative key array arrangements;

Fig. 13 shows an example table of associatively linked and prioritised data strings; and

Fig. 14 illustrates the chaining of data dictionaries and associative linking.

The present invention is directed to an efficient (productivity), facile (accessibility) and safe (ergonomic) keyboard for single and dual hand, full and limited dexterity, and right or left hand orientation users as a Multi-Dexterous Productivity keyboard system, which among its aims includes: (i) the effective reduction of key-stroking/typing, thereby (ii) increasing efficiency (productivity), (iii) increasing ease of use (accessibility), (iv) increasing safety (ergonomics), and (v) reducing ailments associated with keyboard use.

Fig. 1 shows a conventional keyboard according to the QWERTY layout standard. The keys are arranged in straight rows with a user's hands shown to illustrate the natural position of the fingers in a relaxed typing position. The tips of the fingers form a natural arc with respect to the keyboard by virtue of the differing lengths of the fingers and

thumb of each respective hand. To conform to straight rows of keys of the key board, fingers are forced to be held in an unnatural position while poised over the row of conventionally designated "home keys". This unnatural position causes significant hand discomfort from repetitive key strikes and makes touch-typing more difficult due to the tendency of the fingers to stray or extend from the home row of keys. Thus, the conventional straight line of home keys is a source of ulnar deviation and pronation both of which are causes of Repetitive Strain Injuries (RSI) for regular keyboard users.

A basic keyboard of a personal computer, whether physical or graphically represented, can include further keys that permit a direct reduction in a user's physical interactivity with the device using the fundamentals of etymology. These additional keys provide a means to input diverse patterns based on language or graphics and represent particular lexical fragments or basic components of such languages or graphic systems.

Core lexical components or data string fragments combine to create larger data strings. The phrase "data string" and "character string" are interchangeable throughout the specification unless the context requires otherwise. Similarly, depending upon the context, the term "sub-data string" or "truncated data string" may refer to letters or lexical fragments within a word, or a

word within a phrase or sentence, mnemonics, abbreviations, acronyms etc.

For any given language, its core lexical components (letters, numbers and symbols) and its most occurring character string fragments can be used to create larger complete character strings that become contextual by representing meaningful words, phrases, sentences, paragraphs and fuller texts. Such patterns can include the most frequently occurring digraphs (two-letter combinations forming a single lexical unit, e.g. TH, ER, EN, AN etc.), tri-graphs (three-letter combinations forming a single lexical unit, e.g. ENT, LLY, TCH, ATE etc.), tetra-graphs (four or more letter combinations forming a single lexical unit, e.g. TIVE, ALLY, MENT, ENCE etc.) and sym-graphs (emoticons, e.g. :-) for smiley etc.). The same principles apply to graphic systems by using common and simpler abstract patterns to generate larger, more complex graphic patterns. Those fundamental components occurring with the most frequency in any given language are most useful as key legends or indicia.

The lower the length or size of these core lexical components, the greater their simplicity and the more amplified their cognitive coherence. Cognitive coherence measures a character string's diversity, versatility and breadth of contextualisation in terms of reusability and/or its ability to build larger character strings easily and repeatedly. Letters, numbers and symbols have the highest

cognitive coherence since they represent the basic lexical/numerical components and building blocks for any given language. Words, phrases, sentences and fuller texts have lower cognitive coherences the higher one goes up this chain. Digraphs have a particularly high cognitive coherence since they are practically at the bottom of the chain, having a similar cognitive coherence to that of letters. Digraphs can be loosely coupled with other letters and patterns to create larger more meaningful character strings, semantics and contexts.

The use of digraphs, tri-graphs and tetra-graphs provide easy acclimatisation toward their use because of their high cognitive coherences; i.e. they are easily recognisable and easy to place within larger patterns during the construction of meaningful words, phrases, sentences and fuller texts within any context or semantics. Digraphs, tri-graphs and tetra-graphs also reduce the amount of physical interactivity by facilitating a reduction in the number of key presses required to create a given character string. This may be achieved by eliminating key-presses by means of providing data input keys (either physical or graphically represented) having multi-character indicia which correspond with a statistical extrapolation of the most used alphanumerical character combinations (i.e. letters, numbers and symbols) in a given language by the user.

Modified QWERTY keyboards according to the present invention are shown in Figs. 2a and 2b.

Advantages of the keyboards of the present invention include ease of use, reduced user-interactivity, elevated efficiency and thus enhanced productivity that in turn yields improved accuracy and flexibility. Reduced interactivity is a stress/strain antidote that reduces the risk and occurrence of Repetitive Strain Injuries (RSI). Furthermore, reduced interactivity has the further benefit of lessening wear and tear of the personal computer itself. Comfort is a palliative benefit. The only effective way to improve ergonomics and prevent injury is to do less of any activity, e.g. reducing the number of key presses involved in typing.

The keyboards and interface system of the present invention improve the overall user experience and interactivity with a personal computer. The apparatus can be used independently of the interface system that forms a first aspect of the present invention (described in detail below), or for maximum benefit, both the keyboards having keys with multi-character indicia and the interface system may be used in combination.

The keyboards of the present invention are arranged with a particular symmetry that enables them to be easily split into three segments (as shown in Fig. 2b) to provide greater flexibility in approach and

comfort, thus further enhancing ergonomics. Here, the first split would tend to be between the central two columns of the keys having multi-character indicia (described in detail below) and the other split would tend to be between the main keyboard section and the numeric/calculator section. This applies to all configurations of the keyboard according to the present invention.

The keyboard comprises an array of keys having multi-character indicia, shown in isolation in Fig. 2c, arranged substantially centrally between the home keys F and J. The keys within the array are composite keys having at least primary and secondary indicia arranged such that the most frequently used multi-character combinations in a given language are positioned closest to the home keys.

It will be appreciated that the keys having multi-character indicia could equally be provided on a DVORAK or MALTRON® keyboard or on an array of keys represented on a dynamically updated graphical touch screen which repositions the multi-character combinations keyed most frequently by a user such that they are positioned closest to the home keys.

Each key within the array of keys having multi-character indicia (hereinafter referred to as MCI keys) has primary and secondary functional indicia disposed on its top surface wherein at least the primary functional indicia is statistically

extrapolated (discussed below in more detail with reference to Figs. 5a-i).

The primary form of functional indicia on each composite productivity key shown in Figs. 2a and 2b are in the form of digraphs. However, it will be appreciated that other forms of primary functional indicia are possible and may be in the form of at least one of: characters (single letters, tri-graphs, tetra-graphs), words, word groups and/or special commands all of which serve to alleviate the recognised problem of repetitive key strikes and/or alleviate excessive redundancy, repetitive typing and/or optimise typing productivity based on the most commonly used characters, words, word groups and special functional commands of any given language including (for example, English by default, French, German, Italian Spanish and other EU and international languages).

Each digraph is selected using the results of a statistical data study of the most commonly used words in the English language. The statistical data study has shown that the following digraphs (i.e. coupled letters) occur most commonly in the English language: OF, OR, IN, EN, ES, RE, TH, AT, ED, ER, ON and AN. In view of the fact that the Q key is rarely used singularly (according to the statistical studies discussed below) but is often paired with the letter U, a QU digraph key is provided. However, since this digraph is less common than the others are, it is not included in the central

productivity key array and retains the position of the conventional Q key.

In an alternative embodiment (not shown), each set of characters, words or word groups are taken from a statistical data study of the most commonly used tri-graphs (i.e. three-character combinations) such as QUE, QUA, QUI, THE, ETH, ITH, ION, ONE, TEN, ENT, END, ENV, FOR, TOR, TER, FER, GER, BER, INT, INY, REY, REG, GED, EDY, AND, ANY, ANI, etc. Experimentation has shown that the use of productivity keys using digraphs and tri-graphs can reduce multiple keystrokes by up to approximately 30%.

Furthermore, the composite productivity keys shown in Figs. 2a and 2b have secondary indicia of the most used special software application based commands, acronyms and/or mnemonics, by default Internet Top Level Domains (TLD) (i.e. ".tv", ".info", ".org", ".edu", ".gov", ".mil", "www.", ".co", ".ac", ".ccode", ".net" and ".com").

These TLD's are all well known with the exception of the ".ccode" TLD. This secondary key value is user definable during the keyboard driver installation or run-time configuration to correspond with the most commonly used top-level domain (TLD) value. For example, if the keyboard is to be used in the United Kingdom, a user would select the United Kingdom from a list (as shown in Fig. 3) during installation or

run-time configuration of the keyboard driver thus assigning the value ".uk" to the .ccode key.

The MCI keys are configured to have default linguistic settings that are function key controlled. The primary and secondary productivity key-values are programmable during installation or run-time configuration tools. Whilst in normal mode, the default key-values of the productivity keys shown in Fig. 2c will be the digraph values. For example, pressing "EN" alone will give "en". Pressing the "SHIFT" function key in combination with key "EN" will produce "EN" in upper case. In Caps lock mode the values summoned would be "EN" and "en" respectively. Further composite keys include DUAL, which accesses secondary key values, and DUAL SHIFT which accesses and shifts on secondary key values. In normal mode the "DUAL" key used in combination with key "EN" summons ".edu" and "DUAL SHIFT" summons ".EDU". Further examples incorporating usage of the .ccode key are shown in Fig. 4.

In the particular example shown in Figs. 2a, the productivity keys are arranged in substantially the central area of the keyboard in an array comprising two columns, which intersect with two rows in a substantially mutually perpendicular arrangement, each row and column consisting of four productivity keys. The two rows lie adjacent to one another such that the first and fourth keys of the first and second rows intersect with the second and third keys

of each column respectively to form an H-shaped array. In other words the first and fourth keys of the first and second rows are shared with the second and third keys of each column.

The H-shaped array means that a single or dual handed user has immediate access to the most commonly used characters and commands at his/her fingertips without unnecessary flexing or extensions beyond conventionally designated home keys. The most used or most frequently occurring MCI keys (derived from any given language statistical extrapolations or of general information) are positioned in order closest to the keyboard home keys F and J. In addition, a user is not presented with the drastic psychological factors of having to relearn how to use an unfamiliar style of keyboard since the familiarity of the QWERTY model is retained and merely modified to improve efficiency and to reduce repetitive key strikes and the like to minimise Repetitive Strain Injuries (RSI).

Additional function keys which are operable in combination with the productivity keys (i.e. the BSPC (Backspace), DEL (Delete), DUAL and DUAL SHIFT keys are added to the array as shown more clearly in Fig. 2c to form an overall array comprising 18 keys (i.e. 4 function keys and 14 productivity keys - excluding the generic QU key but including the <space>T and E<space> keys described below).

The BSPC and DEL keys are well understood and require no further explanation and the DUAL and DUAL SHIFT keys have been described above. However, the "SPC T" and "E SPC" (i.e. <space>T and E<space>) keys are new keys, which contribute to increased typing efficiency. A statistical analysis of the English language has shown that the most common characters that start and end a word in the English language are the letters "T" and "E" respectively (see Figs. 5b and 5c, respectively). Space (SPC) delimits and/or indicates the start or end of a new word or a previous word respectively. Accordingly, these keys serve to provide a reduction in keystrokes in a similar manner to digraphs and so are notionally included within the group of MCI keys. These keys have secondary key values .biz and .pro respectively that are accessible using the DUAL keys as described above.

Further functional keys (not shown in the example of Figs. 2a-c) may also be added to the array. For example, "iBusiness" and "iPersonal" keys are programmable keys via the keyboard driver (during pre and post driver installation). In an alternative example (not shown) these keys replace the BSPC and DEL keys located at the top of the array shown in Figs. 2-c. The value of the iBusiness key is defaulted to the user's business web site, e.g. www.keypoint-tech.com. The value of the iPersonal key is user-definable and is intended to default to a user's internet home URL setting. During installation or run-time configuration of the

keyboard driver these defaults can be amended by the user. URL validation will be made to verify the correctness of the URL format and page access (i.e. the URL is ping-ed). Changing the iPersonal key value will not affect the user's pre-existing internet home URL setting as these are maintained independently of one another. In operation, the keyboard driver will therefore either feed the selected URL value into the internet browser address field (or into any cursor area during cursor/text input mode), or auto-start-up a browser with the selected URL when not in cursor/text input mode.

Yet another pair of functional keys (again not shown in the example of Figs. 2a-c) can be added to the array. These are the "MULTI DEL" and "MULTI BSPC" keys respectively. Again, these keys contribute towards a reduction in keystrokes by deleting n characters, words, sentences or paragraphs at a time either from left-to-right with MULTI DEL or right-to-left with MULTI BSPC. The user can associate n to characters, words, sentences or paragraphs during keyboard driver installation or run-time configuration.

Optionally, it is envisaged that the MCI keys of the first embodiment could be provided with graphically programmable liquid crystal display (LCD) key-tops (or a touch screen) which are dynamically programmable in real time. The keyboard driver would be adapted to have a two-way channel that dynamically programs the indicia of the keys, or

touch screen representations thereof, in real time according to the current software application in use. Alternatively, the user can configure the keys by selecting which indicia should be attributed to each LCD key-top.

It will be appreciated by those skilled in the art that the functionality and layout of the MCI keys will minimise Repetitive Stress Injuries (RSI) such as Carpel Tunnel Syndrome (CTS) and other Cumulative Trauma Disorders (CTD) such as Musculoskeletal Disorders (MSD), Occupational Overuse Syndrome (OOS), Repetitive Motion Injury (RMI), Upper Limb Disorder (ULD), etc. in dual or single handed keyboard users, full or limited dexterity keyboard users and left or right hand oriented keyboard users. The primary difference being that a user now has an optimally arranged set of keys formed with statistically extrapolated indicia or characters and special commands that significantly reduces unnecessary finger extensions and related fatigues beyond a user's hand span. Additionally, workload is reduced thereby reducing or pre-empting stress and/or strain.

Since each language whether English, French, German, etc. has distinct linguistic characteristics inherent to its etymology and principal area of technological or otherwise application of origin, it would be obvious to one skilled in that language to construct special primary commands to provide the necessary functions and language based commands.

In this regard, numerous key orientations are possible, excluding those orientations that minimise typing speed, and detract from preventing repetitive key strikes that can lead to Repetitive Strain Injuries (RSI), and potentially decrease efficiency (productivity). The tables in Fig. 8 show examples of statistical extrapolations of the most commonly occurring language components for the English, French, German, Italian and Spanish languages.

Other defining benefits for other users can be derived by rearranging the inherent factors: key-presses, effort, dexterity, and time that measure efficiency, effectivity and accuracy. This simple exercise provides the foundations to reap 'whatever advantage for who ever'. For example, (i) Time: Military / Critical-path systems; (ii) Productivity: Commercial /Customer services; (iii) Accessibility: Governments / People with disabilities; (iv) Accuracy: Health & Legal / Emergency services; (v) Ergonomics: Trade Unions / Workers injuries; (vi) Growth: Education / Future Markets; and (vii) Change: R&D /Product Diversification etc.

A closer look at statistically extrapolated character, word and/or command data or indicia is shown in Figs. 5a-i. Through the analysis and weighting of the most common usage in English, lexical fragments such as combinations or subsets of letters, digraphs, tri-graphs and small words can be extracted. The exclusive union of these categories (frequency, union), filter out duplications of 1 to

3 letters from each respective subset resulting in a compacted optimal mix of combinations that can be used in keyboard design applications to recreate fuller words thereby minimising repetitive keystrokes and associated injuries. For instance in Figs. 5a-i, it is shown that the most popular combination of digraph is TH, for tri-graph it is THE and the most used letter is E. The succinct union of these combinations, in turn are process similarly with other unions and the final remaining contents (superset) listed as the most likely candidates that can be use to reduce key strikes or strokes, repetition and key reaches.

As depicted in Fig. 5a, a table generated from a variety of studies shows the most common weight of letters and there frequency of use. As shown therein the letter E had the highest frequency as the most used letter in the distribution of data. As depicted in Fig. 5b, the table shows the letter T as having the highest percentage frequency of most used letters that start a word. As depicted in Fig. 5c, the table shows that the letter E as having the highest percentage frequency of the most used letters that end a word in English. In order to generate the most effective union of the selective data, a criteria is imposed to systematically eliminate the less frequent letters and leave only the most popular ones. These in turn, are used in the final selection and optimisation of a superset. This can be viewed more clearly in the combinations obtained from digraphs and tetra-graphs depicted in

Figs. 5d and 5e. As shown in the table of Fig. 5e, the three letter word THE is shown as having the highest percentage frequency with AND as next likely candidate for selective combinations. As indicated by the table selecting the most prominent and primary tri-graphs with frequency values greater than 6.10, along with a secondary set with frequency values between 5.00 and 6.10 optimum tri-graph sets can be obtain (e.g. primary set: AND, ENT, FOR, ION, THE, TIO; and secondary set: EDT, HAS).

As depicted in Fig. 5f, the table shows the percentage frequency of the most used words of the English language as the sample set wherein words such as AND, IN, OF, THAT, THE, and TO were optimally obtained imposing the criteria of frequency values of greater than 0.9 to obtained the discrete word set. The percentage frequency of words that are two or more letters in length strongly indicates that the bulk of keyboard operations require repetition and multi-strokes. As depicted in Fig. 5g, the recorded data shows a distribution curve that indicates that the main weight in frequency is consumed by words of two to five letters in length, which represents 74.17% of a possible workload. The introduction of the optimised frequency union from letters, digraphs, tri-graphs and small words significantly reduces this workload by simply eliminating unnecessary typing. Accordingly, this inherently reduces the risks associated with the duration of keyboard operations and subsequently diminishing Repetitive

Strain Injuries (RSI) and elevating productivity.

As depicted in Figs. 5h and 5i, keystroke reduction is shown for the most common tri-graphs and most common words, respectively. The select combination from the frequency union also includes OF, OR, IN, EN, ES, RE, TH, AT, ED, ER, ON and AN. For example, the use of the tri-graph AND indicates a keystroke reduction of 14.55 keystrokes out of a sample of 100 words. Using the most common word groups the same word AND indicates a greater keystroke reduction of 28.30 keystrokes out of a sample of 100 words.

Figs. 6a and 6b illustrate the composite key operations for various key value combinations of a specific set of data in normal typing mode and utilising the function keys SHIFT, DUAL and DUAL SHIFT under Normal and Caps Lock mode. As set forth in Fig. 6a, composite key operations performed are tabulated for exemplary letter (t,T), Symbol (3,#), and productivity (th,TH,www.) key sets. These sets represent the default pair of primary and secondary values (and tertiary values for the productivity key), respectively, for their assigned keys in normal typing mode. Note, in capitalisation mode the primary and secondary productivity keys are reversed. The Shift key will work normally as used in existing keyboard operations. The Shift key uses the secondary letter key values. The Dual/Dual Shift keys work only with the secondary or tertiary key values. The pair value for the Q key has been changed and reversed since the letter Q is rarely

used singularly and is primarily paired with letter U to form the majority if not all, fixed QU words. Thus, the qu is the primary value, QU the secondary value and q the tertiary value. For the MCI key data (th,TH,www.), pressing the MCI key in normal mode summons "th", pressing the SHIFT in combination summons "TH", pressing the DUAL key in combination with the productivity key summons "www." and pressing the Dual Shift key in combination with the productivity key summons "WWW." as the normal mode protocol. In caps lock mode, the results for the productivity key data set are respectively the reverse.

As set forth in FIG. 6b, an enhanced data set is shown wherein the Letter, Symbol and MCI key data sets include three assigned values in normal mode, respectively (t,T,the), (qu, QU, q) and (th,TH,www.). Here the influence is the set (t,T,the), which permits normal keys to also have most used word, phrase, abbreviation, mnemonic or command associated with it as a DUAL or DUAL SHIFT accessed key-value. Similarly, function key utilisation according to the above reference protocol can be evaluated by using the key data (t,T,the). In normal mode, pressing the key in normal mode summons "t", pressing the SHIFT in combination summons "T", pressing the DUAL key in combination with the key summons "the" and pressing the DUAL SHIFT key in combination with the key summons "THE" as the normal mode protocol. In caps lock mode, the results for the key data set are also

respectively the reverse. The tables in Figs. 6c-e show full mappings for all other keys.

The primary advantages of the productivity (efficiency) and ergonomic (accessibility) keyboard include the aspects of ergonomics that serve to optimise efficient key access by maximising comfort and minimising unnecessary keystrokes. Inherent to these particular factors includes comfort by retaining the most neutral body positions and by encouraging minimal body movements. The primary focus being to substantially reduce the likelihood or probability of acquiring injuries or disorders by minimising stress and fatigue related various parts and muscle groups of the body such as ligaments, musculoskeletal joints, muscle tendons, hand nerves, and neuromuscular trigger points. In this regard, the invention is directed towards merging the needs of able and disabled persons to provide a keyboard that optimises efficient keyboard use and levels the keyboard playing field to include an added benefit of diminished Repetitive Strain Injuries (RSI) and elevated productivity (increased work throughput).

To this end, it is important to differentiate between the use of a software application and its purpose. The use of the application is defined by the physical operations or functions available via the application interface, keyboard, and pointing device, which help fulfil the purpose of the application. The purpose of an application is its objective to meet user requirements, and to

parameterise its use. Mechanisms that enhance the purpose of applications provide diverse or greater methods of application use. This in turn makes the interface, and its keyboard and pointing device, more efficient and effective.

Conventional keyboards only provide the mechanism to use the applications. The use of the application is determined by the predefined user-permitted operations of the applications that allow it to fulfil its purpose. Thus, conventional keyboards are functional, limited to one-way feeds, from keyboard to application, and do not provide the scope to improve or diversify the application interfaces that would otherwise allow for enhancements to application purpose. For example, in word-processing, all the operations allow one to format and present a document that forms the basis of the applications use. The purpose of the application is to enter text, based on language. Thus, the breakdown of language into its bare lexical components, such as letters, digraphs and tri-graphs etc., would provide a more efficient and easier mechanism to fulfil the purpose of the application. This also permits the application to diversify its functionality and enhances the versatility of what the application can do with its interface. The same principles can be applied to any software application such as financial trading systems, Internet browsers and the like. The ability of the keyboard and interface system of the present invention to enhance not only application

use but also application purpose, via the unique MCI keys, improves user interfaces and permits applications to operate more effectively and efficiently with application diversity and versatility. The ideology of the present invention connects the user with the software applications at the information level, thereby fulfilling more the purpose than the function.

As suggested above, a driver-based approach can be used (either in isolation or in combination with the multi-character indicia aspect of the invention described above) to reduce a user's physical interactivity with a personal computer. Computer keyboard drivers are essential in all operating system (OS) environments, their function being to convert keystrokes to OS language tables, thus bridging or translating required notation within all human-to-computer interfaces. It is important to note that the keyboard driver is a critical element to keyboard function and operation and that the additional features of the keyboard driver of the present invention is also operable with, and can be extended to, all currently available keyboard drivers.

Conventional keyboard drivers merely map key legends to OS language tables with little or nothing in the way of sophisticated extensions or add-ons to improve performance, versatility and adaptability of the keyboard medium.

However, the keyboard driver of the present invention is adapted to implement the enhanced features of the other aspects of the present invention leading to increased typing productivity and keyboard adaptability and versatility. The keyboard driver of the present invention includes a number of optionally activated and configurable modes including a "double-press mode", a "multi-press mode" and a "translator mode" as well as 'mapping mode', 'project mode', 'predict mode', and various attributes governing the control and behavioural aspects of the keyboard driver (operable with a user-configurable dictionary). These modes or features accumulate, grow and maintain all dictionary information, including entry or link statistics, probabilities and analytics (scaling patterns of use: historical usages, contextualization, relative associations and occurrences thereof), as well as dictionary chaining information and various generic counts and operational indicators, which inherently facilitates overall intelligence permitting the MDP to adapt to the user's habits, behaviours and working environment. All modes constitute additional features over conventional keyboard drivers, which results in improvements to typing productivity and adaptability and versatility. The aforementioned modes are described in detail below.

The double-press mode allows a user to select one of two alternative key values/functions depending upon whether a key is pressed once or twice within a

predetermined time period (i.e. similar to the double clicking of a mouse). Normally, the two alternative key values will be: (i) the normal key value (i.e. the value obtained when no function key is used in combination with it); and (ii) the SHIFT value of that key. For example, a single press of key "A" yields key value "a" (i.e. lowercase normal key value) whereas a double-press of key "A" yields key value "A" (i.e. upper case - SHIFT-"A" key value). Of course, the keyboard driver could be adapted such that the second press of a key in double-press mode selects any other alternative key value other than the SHIFT value such as, for example, the DUAL or DUAL SHIFT value.

The multi-press mode is a natural extension of the "double press mode and is activated by two successive key presses within a predetermined period of time to generate a list of appropriate character strings from the dictionaries or by other features that also generate lists of appropriations respective of their determining factors and outcomes, where any list is formulated using entry or link statistics, probabilities and analytics (scaling patterns of use: historical usages, contextualization, relative associations, lengths and occurrences thereof), wherein each progressive press of a relative keyboard key up to n times selects the next appropriate character string from the installed dictionaries or from other derived or given lists of n appropriate character strings respectively. Optionally, to gain any productivity

benefits from the multi-press mode, successive multi-presses must retrieve character strings which: (i) are actively prioritised by frequency of use (either per session or in real time); and (ii) are of a character length greater than or equal to n+1.

For example, a user can toggle between and/or dynamically integrate different user-installed dictionaries. Dictionaries may consist of mobile texting mnemonics, abbreviations, industry specific jargon such as medical acronyms etc. Prioritisation of each word/mnemonic is updated each time it is selected, typed, scanned, or used to event an occurrence thereof. An optional feature is the storage of any new words within the dictionary (with NULL description).

An extension of the multi-press mode is a translation mode, which can be set during installation or run-time configuration to one of the following conditions: OFF, Translate+ (i.e. translate maximise) or Translate- (i.e. translate minimise). When set at Translate+, any character string (for example, a word or mnemonic), whether selected using the multi-press mode or not, will automatically expand into definition/description stored within a dictionary. Alternatively, when set at Translate-, any character string (for example, a phrase or a sentence) will automatically contract into a shortened version (for example, an abbreviation or an acronym) stored within a dictionary. Accordingly, the translation mode

performs two-way translations depending upon the particular settings chosen by the user. Examples of the operation of the translation mode are shown in the table in Fig. 9. As illustrated in the table of Fig. 9, translations can be performed using a variety of configurable dictionaries, which are run-time user configurable or downloadable from the internet in real time.

The dictionaries are used to determine user typing habits and behaviour and adapt the MDP keyboard to the user environment dynamically. This reduces excessive repetition and redundancy within typing, thus further improving productivity (efficiency) and accessibility (ergonomics). The dictionaries hold various statistics (basics include frequency, length, pattern density / versatility, chronological weight and direction / operational indicators etc.), probabilities (basics include cognitive coherence, occurrence ratios, context ratios, and associative index etc.) and run-time analytics (scaling patterns of use: historical usages, contextualization, relative associations and occurrences thereof) that are dynamically updated in real-time and in accordance of use for all entries, links and chains maintained within the dictionaries, and further statistical attributes, software control dynamics, entry / link / chain attributes and indicators may evolve in the future. Many dictionaries can be configured at any one time and each can be of a different type. Duplications are handled by prioritising the installed dictionaries where by

entries within a higher priority dictionary have precedence or are determined if manual overrides have been put in place by the user during installation or run-time configuration. All dictionaries are dynamic and therefore can be duplicated into various other languages or downloaded from the internet and configured during run-time.

The interface system of the present invention (which will be known under the Trade Marks MT-iDICT™ and/or AdapTex™) provides and maintains an adaptive intelligence data dictionary system. This interface system controls and uses various interactivity dynamics, statistics and full descriptions/translations of each entry (e.g. mnemonics, abbreviations or acronyms) stored within one or more data dictionaries installed within a storage means of the MT-iDICT™ interface system. None, one or more than one dictionary can be installed at any given time. Dictionary instalment and configuration thereof can be done in real-time.

Each data dictionary holds qualitative and/or quantitative information relating to a given data string. Examples of qualitative and/or quantitative information are as follows: (i) statistical information relating to a data string's historical usage or selection (i.e. frequency of use/selection, character length, lexical pattern density/versatility, chronological weight and direction/operational indicators etc.); (ii)

probability information relating to a data string's historical usage (i.e. occurrence and/or association ratios of two or more sub-data strings within a longer data string; context ratios determining the likelihood of a given data string being grouped with one or more other sub-data strings to determine the context of a longer data string; or other statistical derivatives based on language and user traits such as timestamp, cognitive coherence, perceptual indices, associative indices, grammar orients, correlative weights, inference ratios and pattern factorisation etc.); (iii) run-time analytics (scaling patterns of use, historical usages, contextualization, relative associations and occurrences thereof); (iv) dictionary priority; (v) dictionary chains (where each chain also retains and uses the information in (i), (ii) and (iii) above); (vi) data string links between other data strings (where each link also retains and uses the information in (i), (ii) and (iii) above); and (vii) translations.

All of the qualitative and quantitative information is dynamically updated in real-time and in accordance of use for all entries or data strings, links and chains, translations maintained within the dictionaries (described in further detail below), and further statistical attributes & software control dynamics.

The data dictionaries can be manually populated. Alternatively, the data dictionaries can be

automatically populated by use of document or text scanners, which scan data strings and assemble their statistics, probabilities, run-time analytics as well as associative links between data strings. The idea being that such documents or texts written by a user reflect the behavioural use of vocabulary and patterns of the language(s) reflected by the user.

A data string may be in the form of a full data string (i.e. a word, phrase, sentence etc.) or a corresponding truncated data string such as a mnemonic, abbreviation or acronym. The prioritisation of data retrieved from a data dictionary is user-configurable to allow a user to prioritise the ordering of data listed on a display means according to selected qualitative and/or quantitative characteristics. The user configurable parameters include system behavioural parameters, data string statistics, probabilities and analytics (scaling patterns of use: historical usages, contextualization, relative associations and occurrences thereof), and dictionary priorities.

In addition to those mentioned above, further qualitative and/or quantitative characteristics may include: (i) the presence or absence of one or more data string fragments in the form of digraphs and/or tri-graphs and/or tetra-graphs etc within a full or truncated data string; (ii) the presence or absence of truncated data strings in the form of mnemonics, abbreviations or acronyms which correspond with the full data string; (iii) two-way translations between

full data strings and their corresponding truncated data strings; (iv) the frequency of two-way verbatim, correlated and/or inferred translations between two languages (i.e. English to French); (v) the character-length of each full data string or its translation or any corresponding truncated data string; (vi) the frequency of selection by a user of each full data string (i.e. words, numbers, symbols, emoticons etc.) or its translation or any corresponding truncated data string; (vii) the frequency of forward and backward translations between full and truncated data strings; and (viii) the frequency of forward and backward verbatim, correlated and/or inferred translations between two languages. Each data dictionary may also hold indicator flags that dictate and delimit control and use of the stored data by the software, and the level that it pertains to relative software tiers.

Data strings stored within the data dictionaries are selected/accessed using the first character of the data string, and ordered by descending frequency and ascending length for basic default sequencing. The ordering is configurable by the user using any field (qualitative or quantitative) of the data dictionary. Ordering can also be configured to be ascending or descending. The first character is sourced from a single key press or a composite group of first characters obtained from key presses.

A configuration tool permits setting the various behavioural aspects (also known as physical

interactivity reduction characteristics) of the MT-iDICT™ interface system. The behavioural aspects (physical interactivity reduction characteristics) are as follows: (i) automatically entering a space after a selected full or truncated data string; (ii) limitation of displayed mnemonics to those having a total number of characters greater than the number of key presses required to display said mnemonic on the data display means; and (iii) automatically performing forward or backward translations between mnemonics or abbreviations or acronyms and their corresponding full data strings.

Further behavioural aspects include specifying the number of selected entries to be displayed or listed on the display means at any one time, maximising a mnemonic to become the most frequent of its category with highest priority, editing of entries, or ordering run-time selections based on certain qualitative or quantitative characteristics in ascending or descending order etc.

Further behavioural aspects include specifying a projection of n words or sentences by way of using the associative indices and other qualitative/quantitative statistical derivatives.

The interface system can also determine a user's most frequently used phrases (i.e. full data strings) and automatically abbreviate or implode them into a mnemonic, acronym or other abbreviation (i.e. a truncated data string). This allows a user

to have fewer key presses via the truncated data string which can then be manually or auto-translated into its corresponding full data string. See the Trans+ and Trans- buttons on the personal computer of Fig. 2a which can be used to perform manual imploding or exploding of data strings. The personal computer can also be configured to perform this function automatically.

A limited number of most used entries pertaining to a key press can be displayed at any one time. Additional entries can be scrolled through using the navigation means up to a maximum set by the configuration tool.

The diversity of dictionary types is enormous, e.g. one thousand most used words, mnemonics, acronyms, abbreviations, conversions, Short Message Service (SMS) texting data, emoticons or other data specific to the user and/or a user's working environment etc. Data dictionaries can be even more specialised by being departmentalised within specific working environments. For example, in a medical environment the dictionaries can reflect symptoms and remedies, ailments and pharmaceuticals, or simply provide normal medical terms and their definitions. In a reservation environment, the dictionaries can reflect airlines, destinations, flight codes, seating, hotels, prices etc. In an investment trading environment the dictionaries can reflect trading instruments, traders, portfolios, Reuters Instrument Codes (RIC), trader specific RICs,

quantities, buy/sell prices and forecast analytics etc.

Dictionaries can also be integrated into any other software and controlled dynamically to reflect changing circumstances to the entries within respective dictionaries. This provides real-time adaptive intelligence relative to the user, working environment and type of system being used adjacent to its purpose.

The real-time maintenance of dictionaries and the dynamics of the MT-iDICT™ interface system allow it to contour towards a user's traits and uses of the device, along with the user's use of language and level of vocabulary. This enables the MT-iDICT™ interface system to be adaptive and intelligent relative to the user's volume, level and type of use of the system. Over time, the data dictionaries will evolve to reflect the most favourable and most appropriate or relevant data strings used by the user and thus adapt and contour the MT-iDICT™ interface system relative to, and more appropriately towards, the user.

As with the multi-character indicia aspect of the present invention, the benefits of the interface system include ease of use, reduced user-interactivity, elevated efficiency and thus enhanced productivity that in turn yields improved accuracy and flexibility. Reduced interactivity is a stress/strain antidote that reduces the risk and

occurrence of Repetitive Strain Injuries (RSI). Furthermore, reduced interactivity has the further benefit of lessening wear and tear of the personal computer itself. The combination of both the multi-character indicia aspect of the invention together with the software elements of the interface system provides the greatest benefits in terms of facilitating a reduction in the number of physical key presses required to create a given data string.

Synchronisation of users' data dictionaries between personal computers maintains accurate translations, semantics and meanings. Synchronisation can occur or be accomplished using infrared, Bluetooth® or other wireless connectivity methods available on personal computers. Alternatively, central repositories or databases can be maintained by the communications service providers that computers can access easily, or they can be maintained and accessed/downloaded via internet web sites. These synchronisation mechanisms maintain consistency of dictionaries and their use thereof by groups of users. The central repositories (i.e. internet databases) provide a means to standardise dictionaries for the general population of users.

Once the interface system software and MT-iDICT™ data dictionary facilities are integrated/installed into the computer, the software aspects can use and process MT-iDICT™ data dictionaries using standard systemic logic to achieve optimum utilisation, i.e. using best processing methods and techniques to

obtain all the efficiency benefits. The configuration tool also permits the scanning of existing messages resident on the personal computer or the downloading/transfer into the computer (i.e. by either Internet, another PC or other compatible device using cable or wireless technologies) of dictionary data in order to acclimatise the MT-iDICT™ data dictionaries relative to the data strings used within the messages.

The interface system software uses the MT-iDICT™ dictionaries according to the key press sequences entered by the user either in passive mode or in active real-time dynamic mode. Various navigation features can be used in parallel or adjacent to the interface system software in order to rapidly access the most frequently used (i.e. typed) information. The interface system software reduces the physical aspects of repetitive and recursive typing thereby enhancing efficiency and ease of use and improving the overall effectivity and experience in using a personal computer.

The present invention includes various physical interactivity reduction features (PIRS) which facilitate a reduction in the number of key presses required to create or delete a given data string.

Screen options or existing physical buttons can be used to perform translations (see the 'Trans+' and 'Trans-' buttons in Figs. 2a and 2b). Double pressing of a given key on the keyboard accesses the

most used word or phrase beginning with the tapped letter or generates a prioritised list of the most frequently used words corresponding to the typed letter or letters. This allows the user to conveniently select the desired word or phrase from the list. Alternatively, double pressing can be configured to simply upper case the typed letter or letters.

The first data string or system option in a Pop-Up Selection List (PSL) is highlighted for selection by the user by default. The highlighted data string or system option is selected/activated by scrolling or using other cursor navigation controls. Highlighted data strings or system options are also automatically selected if any other key is pressed, or via a navigation movement.

Alternatively, the first letter of each data string is underlined whereby pressing the relevant key selects the data string or system option without the need to scroll or navigate to it first. Where there is more than one data string or system option with the same initial character, these are scrolled through in the order presented in the Pop-Up Selection List (PSL).

When the PSL is displayed, the desired data string (for example, a mnemonic) can simply be selected by directly scrolling to it. Alternatively, if the desired data string does not appear in the list, the

next letter of the data string is typed to further filter the PSL.

The MT-iDICT™ interface system is not a Predictive Typing Systems (PTS). PTS integration with MT-iDICT™ interface system would allow the PTS to predict more accurately since it is adapting to the users vocabulary in real-time and can presume to hit the users most used data strings (i.e. words, mnemonics, sentences etc.) at every instance.

The MT-iDICT™ interface system formulates logic and prioritisations derived from the data storage qualitative or quantitative information, methods, frequencies and patterns of behaviour and usages of words/mnemonics of the user. Thus, it becomes adaptive to the user and the user's vocabulary and traits. This provides the most favourable and most appropriate and relevant choices for the user based on the user's actual vocabulary, historic usages, frequencies, patterns of use, methods and prioritisations, each being derived from the qualitative or quantitative information stored in the data storage means. The MT-iDICT™ interface system provides data string choices based on actual usages rather than on guesswork as to what the user is trying to create relative to a static generic dictionary.

Predicting typing systems do not reduce the amount of interactivity as effectively as MT-iDICT™ interface system purely because the former still

requires further key-presses to guide its prediction, whereas the latter simply provides discrete choices of full or partial data strings (i.e. shortcuts, whole words, phrases, or partial data strings that can be used to build up or complete fuller data strings, e.g. digraphs, tri-graphs, tetra-graphs and symbol-graphs).

When the personal computer is in text input mode, PSL's are displayed upon detection of an activating tap and/or appropriate navigations by the user. The PSL's show the most frequently used or most appropriate or relevant data strings for each letter or digit associated with the pressed key.

User typed data strings are entered into the MT-iDICT™ dictionary when no such entry exists. This mechanism allows the device to adapt to a user's usage and a user's environment that dictates the type and level of use. The new entries are immediately accessible by the MT-iDICT™ interface system. Thus, the MT-iDICT™ adapts dynamically in real-time making interaction for the user more familiar and making relative information more apparent to use and/or access.

Software facilities, inserts or application macros can be accessed using the PSL facility also.

An extension to the translation mode is the automatic generation of acronyms, abbreviations and conversions. Here the keyboard driver can

dynamically determine acronyms, abbreviations and conversions for such linked associations, thereby providing automated translator shortcuts for the most recurring or commonly used phrases, sentences or texts of n character strings, which can be stored and maintained within any dictionary and made readily available. The user is made aware of such automated acronyms, abbreviations and conversions via the keyboard driver dictionary console, display / reporting and edit features where the user can also create personalised shortcuts and where these shortcuts can include system or device commands and executable instructions / macros.

The interface system is also provided with a 'mapping mode'. Dependent on this mode being activated and various chains between dictionaries being predefined and established by the user during installation or via run-time configuration tools, or automatic chaining being activated, the keyboard driver will perform chained translations of typed or highlighted text. This involves the keyboard driver scanning and mapping appropriate translations from one dictionary to another. Here the keyboard driver maintains lookup chains between any dictionaries such that dynamic mapping can be made from one dictionary to another, and so on. For example, English-to-French (dog, chien) and French-to-German (chien, hund) dictionaries can be chained such that it can infer English-to-German (dog, hund) mapping.

More sophisticated dynamic mappings could chain, for example, a symptoms dictionary to a prescriptions dictionary whereby relevant character strings are matched between any dictionary entries and translations to dynamically chain such dictionaries together and induce n ailment to medicine mappings. A single mapping is definitive whereas a list of n mappings are prioritised accordingly and made available via the PSL feature. The number of chained dictionaries is dependent on the number and permutations of installed dictionaries.

The interface system is also provided with a 'project mode'. When activated, the various links between entries within respective installed dictionaries (the links being predefined or established automatically or manually by the user during installation or run-time) allow the keyboard driver to determine and project the most likely associations between n entries relative to the typed or highlighted text. The most relevant, user contoured and adaptive appropriations spanning n derived sub-data strings are then displayed for selection by a user.

The keyboard driver maintains associative links between data strings within two or more dictionaries, such that these links can be used to dynamically infer relative associations between data strings based on link statistics, probabilities and analytics (scaling patterns of use: historical usages, contextualization, relative associations and

occurrences thereof). This allows the keyboard driver to project and retrieve the n most likely appropriations or closely associated data strings from the dictionaries that are relevant, contextual, definitive and user oriented, and each data string being relative to a previous linked association or typed data string.

Optionally, the interface system can dynamically retrieve a list of alternative appropriations relative to each linked association used to induce each of the n respective data strings, whereby each list of alternative appropriations are prioritised and made available via the PSL feature. Once a longer data string is selected from the PSL, this dynamically induces and propagates a further relative projection and retrieval of n further data strings, each being relative to a previous linked association or multi-press selection.

In predict mode, dependent on this mode being activated, the keyboard driver will derive a best match or appropriation relative to the current typed letter or letters, or typed or given / highlighted text pattern, where these letter, letters and/or patterns are the initial letter, letters and/or patterns of entries selected from the dictionaries. Here, the keyboard driver can dynamically best match current typed letter or letters against most likely appropriations from the dictionaries, where the appropriations all begin with the same typed letter or letters. These appropriations are dynamically

retrieved based on given priority of each installed dictionary and entry statistics, probabilities and analytics (scaling patterns of use: historical usages, contextualization, relative associations and occurrences thereof). A single best-matched appropriation is definitive where as a list of best matched appropriations are prioritized accordingly and made available via the multi-press feature.

Fig. 13 shows a table of associatively linked and prioritised data strings. The MT-iDICT™ interface system can multi-link dictionary entries to other entries within the same and/or other dictionaries. These links are based on analytics of patterns of use or relativity between the linked entries. These analytics are dynamic because they change priorities and switch context according to patterns of use.

Thus, a user can specify n projections whereby MT-iDICT™ will link entries to give n sequential appropriation lists of up to, say, five subsequent outcomes relative to a previous entry. Each subsequent appropriation list is prioritised and each can then be selected out of the five if required, most likely not since the top entry for each list will be most likely for use.

For example, if the word "Next" is typed then the projected words (sub-data strings) shown in Fig. 13 would appear (i.e. 'generation', 'of', 'adaptive', 'intelligence', 'interfaces'). Each projected word produces a PSL (for example, the word 'generation'

produces a list of other words below it) that can be scrolled or otherwise navigated through for selection when a user skips to each projection unless a user accepts the suggested projection. The PSL is in priority order of patterns of use and context switching. The spacing in the table is for clarity only and would not appear on as such on the display.

If a suggested word is altered then the subsequent words would change dynamically, contextually as well as associatively and relative to the new selected word. The user can alternatively type a new word from scratch over any original word selection.

On typing each letter of the word 'Next', appropriate selection lists are derived where the beginning of each list entry reflects the current typed letters. For example, typing the letters 'Ne' would provide a list of say, 'Next, Never, Neither, Neighbour, Nederland'. From such a list the highest weighted entry would be shown, in this particular example 'Next' and the letters 'xt' would be highlighted and available for selection to complete word 'Next'.

The MT-iDICT™ interface system will also appropriate the word as it is typed and dynamically change the projections according to any changes to it. This makes it much more adaptive intelligent than it already is. It is envisaged that future versions will have true syntax, context, semantic and grammar

projections derived from LONGMANS, WEBSTERS, COLLINS and OXFORD research dictionary data. An option to highlight only words within a projection that require changing is provided, where remaining unselected words are not dynamically changed.

Continual flow from one selection to subsequent words should be provided such that a SPACE or cursor movement is adequate to perform a selection without the need to use additional select methods, i.e. a cursor movement from a highlight auto-selects the highlighted item unless another mechanism is used to do otherwise.

Optionally, the data processing means can provide manual or automatic spell check features.
Optionally, the data processing means can provide a freeze point enabling the retrieval of static constant appropriations as opposed to dynamic, and which can be based on either most recent or current captured entry statistics, probabilities and analytics (scaling patterns of use: historical usages, contextualization, relative associations and occurrences thereof), or manually intervened prioritisation or overrides.

Duplications are handled by prioritising the installed dictionaries whereby entries within a higher priority dictionary have precedence or are determined if manual overrides have been put in place by the user during installation or run-time configuration.

The following provides an example method to represent dictionary information, indexing and chaining as shown in Fig. 14. It also depicts an example method to represent dictionary entry information, indexing and linking. Although RDBMS could be used, a dynamic method could use system character code tables or repertoires that come in standard ASCII, ISO and other formats that also include language character variants. The system character codes provide the index to each series of dictionary entries that begin with that code. Subsequent entries of the same code are dynamically generated and linked to the previous entry in the same array for that code. Each entry holds its own statistical derivatives such as (i) timestamp (the date and time the dictionary entry, chain or link was created, last used or accessed; (ii) translation, expansion, frequency, length, cognitive coherence (i.e. measures the versatility & flexibility of patterns, their ease of re-usability and the placement of language based patterns); (iii) perceptual indices (i.e. measures the strength of recognizing patterns and deciphering language based patterns, even when patterns are incorrect or misspelled); (iv) associative indices (i.e. measures the relativity between two or more patterns by calculating the frequency of a combination of words or the relationship between words); (v) grammar orients (i.e. the lexical syntax or placement of patterns and their semantics and contextual positioning of nouns, verbs, adverbs, adjectives

etc.); (vi) correlative weights (i.e. measures the semantic relationship or association between two or more patterns where different words mean the same or elaborate other words similar to thesaurus weights); (vii) inference ratios (i.e. measures the likelihood of a semantic relationship or association between two or more patterns by assessing the occurrence of one word or sub-data string within a grouping of other words or longer data strings; (viii) pattern factorisation (i.e. measures the ability to create/breakdown larger patterns from/to smaller patterns wherein in a textual or graphic context, letters, numbers & symbols have highest factorization, then digraphs, tri-graphs, tetra-graphs, 5+ letter words, phrases, sentences, paragraphs, chapters and finally whole texts in this order of factorization).

Additionally entry links are formed to associate entries between themselves, i.e. maintain etymological relationships and statistical derivatives between entries. These entry links again are indexed using system character code tables. The system character codes provide the index to each series of entry links that begin with that code. Subsequent links of the same code are dynamically generated and linked to the previous link in the same array of that code.

Dictionary chaining provides correlation and inference between dictionaries and their entries and

links. Entry linking provides inference and association between entries and their links.

This method allows dynamic generation of dictionaries and their variable entries and respective entry links. It also provides an example indexing system for rapid access to entries and their associated or related link entries. The method permits a spatial/multi-dimensional matrix to represent dictionary dynamics.

A unique aspect of the present invention is that it provides mechanisms or Application Programming Interfaces (API) that allows other software systems to utilise and benefit from all the features of this invention, and to enable improved experiences for the user with such software systems. Additional to this, the API allows other software systems data storage or information repositories to be handled by this invention in similar manner to its own dictionaries etc.

Predictive Typing Systems (PTS) do not reduce the amount of interactivity as effectively as the MT-iDICT™ interface system purely because the former still requires further key-presses to guide its prediction, whereas the latter simply provides discrete choices of full or partial data strings (i.e. shortcuts, whole words, phrases, or partial data strings that can be used to build up or complete fuller data strings, e.g. digraphs, tri-graphs, tetra-graphs and symbol-graphs).

The MT-iDICT™ interface system has a standardised 1000 Most Used Words, SMS text entries, acronyms, conversions and emoticons. However, additional dictionaries can be installed as standard either when the personal computer is shipped or when users pre-install their bespoke dictionaries on setup. These new entries can be edited by the user at will.

The examples shown in Fig. 13 assume that the MT-iDICT™ interface system is in static mode, whereby the sequence/order of displayed letters associated with their respective key is depicted in conventional chronological order. Whenever in text input mode the illustrated Pop-Up selection lists are displayed according to the activating key and appropriate navigations. The Pop-Up selection lists also depict examples of the most frequently used mnemonics based on prioritizations derived from the data storage (dictionary, data string, chain or link etc.) qualitative and/or quantitative information, methods, and patterns of use or numbers relative to each letter or digit associated with its respective key.

User typed words are entered into the MT-iDICT™ interface system's data dictionary when no such entry existed beforehand. In addition, relevant data string links, associations and contextualization parameters are also derived and maintained for all such new data string entries within the data storage means. This mechanism

allows the device to adapt to the users usage and environment that dictates type and level of use. The new entries are immediately accessible by normal MT-iDICT™ means. Thus, the MT-iDICT™ interface system adapts in real time making interaction for the user more familiar, and relative information more apparent to use and/or access.

Software application names relative to a letter on a key can be set to macro mode, thus when tapping the key the PSL gives option to start an application from its list (e.g. keys W, X, Y, Z: WORD, XCEL, YAHOO, ZANY KONG).

As shown in Fig. 9, dedicated keys or buttons are provided on the keyboard. The multi-press key/button and the Trans mode keys/buttons are used to toggle between activated and deactivates states respectively. Regardless of whether the translation mode is set to OFF, the Translate- and Translate+ buttons allow the user to manually highlight a section of text and press the Translate+ or Translate- key to perform an appropriate translation without overriding the otherwise automatic operation of the translation mode. Indeed manual use of the Translate+ and Translate- keys in this fashion can also be used when the automatic translation mode has been set to OFF.

The MULTI DEL and MULTI BSPC keys (described above) behave slightly differently when used immediately after a translation has occurred. For instance,

when the translation mode is active, the character string "call me asap." automatically expands to "call me as soon as possible." if the character string ends with a non-character SYMBOL (in this case a full stop). The immediate use of the MULTI BSPC key at this point would firstly revert back to "call me asap" before fully functioning as a multiple backspace (i.e. deleting the whole sentence" configured by the user. In the same scenario, a regular backspace key would function as normal and singularly delete characters from right to left.

The keyboard driver also opens a separate installation or run-time configuration window when in multi-press mode giving a range of user-definable options. For example, a user can select the maximum value of n, whereby n is the number of most used words to be retrieved from the dictionary during multi-press mode. A check box is provided to enable or disable the multi-press mode functionality when a key is double pressed only (i.e. without a third press within a predetermined period of time from the second press).

The keyboard driver is of course provided with user-definable speed settings for the double-press and multi-press modes, much like those provided for double-click setting for a mouse. Furthermore, the keyboard driver also provides options for cursor selection in order that a user can visually determine whether or not the double-press or multi-

press modes are active. It will be appreciated by those skilled in the art that the double-press and multi-press modes are particularly beneficial to users having limited use of the fingers.

The MULTI DEL and MULTI BSPC keys can be adapted to operate in the translator mode to successively revert from the stored definition/description of a word to the word itself (i.e. upon a single press) and then delete both the definition/description and the word itself (i.e. upon a second press).

The keyboard driver performs the mapping of keyboard signals, which are buffered on a First-In-First-Out (FIFO) basis. Fig. 11 shows two tables that illustrate the mapping of key press events in a FIFO buffer for the typing sequence "Here's another query" (where underlined letters correspond to the digraphs on the appropriate productivity keys of the first aspect). Although the FIFO buffer will almost always be empty since all key-press events will be mapped and dispatched immediately to the operating system and receptive software application, a temporary buffer to store pending characters is recommended to alleviate any possible operating system of software application delays or latencies or conflicts.

Fig. 11 also shows a schematic operating scenario for a FIFO buffer in "piped multi channel" mode. The multi channel mode operates when the key value FIFO buffer is used simultaneously by two or more

software applications. The multi channel mode will be specifically useful for use with the enhanced keyboard driver of the present invention. The piping of the buffer helps avoid buffer complexity, conflict or contention issues particularly during simultaneous use by two or more software applications.

An alternative multi channel mode can also be implemented by duplicating the key value FIFO buffer thus providing a secondary channel for the input of a second software application. The secondary key-value FIFO buffer is always a dynamic replication of the primary key value FIFO buffer. The primary and currently active keyboard application is the only application that can pop/push/flush the primary key value FIFO buffer. A secondary software application cannot pop/push/flush either key value FIFO buffers. The secondary software application may only feed from the secondary key value FIFO buffer. Such rules ensure that no conflict or contention issues occur regarding the key value FIFO buffer. An alternative is to make the key value FIFO buffer shareable whilst still applying the above rules to give control to the primary software application. All other secondary applications simply feed off the key value FIFO buffer.

Fig. 7 shows a calculator portion of a computer keyboard according to a third aspect of the present invention. The calculator portion of the keyboard according to the present invention has been provided

with a more sophisticated functionality and arrangement.

The keyboard shown in Fig. 7 comprises an array of conventional numerical and calculator operator keys, a plurality of calculator control-keys and a liquid crystal display (LCD) located on the keyboard itself.

The calculator control-keys comprise: (i) a first control key for selectively displaying the results of calculations performed using the array of numerical and calculator operator keys on the LCD; and (ii) a second control key for selectively sending the results of calculations performed using the array of numerical and calculator operator keys to a computer.

The calculator control-keys are operable in combination with the numerical and calculator operator keys to determine (a) whether calculator-related or special characters are displayed on the keyboard's LCD display and/or on an alternative display such as a Visual Display Unit (VDU) via the computer; and (b) whether the results of mathematical calculations performed by the calculator keys are displayed on the keyboard's LCD display and/or on an alternative display such as a Visual Display Unit (VDU) via the computer.

In addition to the numeric key-values 0-9, the functional indicia of the first subset of keys

comprise any or all of the following: +, -, /, *, MR, M+, M-, MC, C/AC, SEND, ENTER, MKUP (Mark Up), %, √ and +/-.

Whereby, the latter four operators are defaults but are programmable during installation or run-time configuration to reflect other standard, financial or scientific mathematical operations.

The calculator control-keys consist of the following two keys each of which can toggle between activated and deactivated states: the "CALC LK" button and the "NUM LK" key. The CALC LK key selectively enables and disables the calculator and numeric keypad functions of the calculator portion of the keyboard. The NUM LOCK key works in the conventional manner whereby when activated it accesses the numeric and operator key values, and when deactivated it accesses the auxiliary key values (i.e. cursor controls, home, Pg Up, Pg Dn, End, Ins and Del). Regardless of the NUM LOCK state, the activation of the LCD calculator via the CALC LK button overrides the NUM LOCK state and localises the numeric keypad to operate with the LCD calculator. The ENTER and SEND keys permit calculations to be localized or relayed to the computer respectively, and both update the LCD accordingly.

When the computer is itself turned off, the CALC LK key can be used to activate or deactivate the keyboard calculator for use as a standalone desktop calculator. This feature does of course rely on the

keyboard having its own battery or solar cell for powering the calculator.

In addition, the calculator is provided with a retention buffer that holds a calculation history of n items including the most recent numeric entries, operators and equated values. It should be noted that the retention buffer is totally separate from the standard calculator memory operated using the conventional memory buttons (i.e. M+, M-, MR, MC). The retention buffer allows a user to scroll through the entries stored in the buffer using the UP and DOWN arrow keys, whereby each scrolled entry is respectively displayed on the LCD display. Such functionality allows the user to regress, recur and/or rectify calculations from any previous point within the buffer. In this way, all new entries from a regressed, recurred and/or rectified point overwrite respective/consequent older entries within the buffer, thus calculations are reciprocated entirely, throughout and accordingly.

Also shown in Fig. 7 are a series of Lock keys (ALT Lock, CTRL Lock, SEQ Lock and DUAL Lock). Selection of the ALT Lock and CTRL Lock keys by a user act as if the conventional ALT and CTRL keys respectively are continually pressed. Similarly, the DUAL Lock key acts as if the DUAL key shown in Figs. 2a-c is continually pressed. The SEQ Lock key allows sequential typing without the need to hold more than one key down simultaneously. Fore example when SEQ

Lock is activated simultaneous pressing of SHIFT and 5 is not necessary to obtain %.

The features of the present invention could equally be incorporated into alternative keyboard styles, for example, the MALTRON® and DVORAK keyboards styles. The keyboard driver is provided with a radio button(s) in order that a user may select toggle between the QWERTY, DVORAK dual-handed, DVORAK left-handed and MALTRON keyboard layouts. Fig. 12 is a table illustrating the key-value mappings between these various keyboard styles.

Modifications and improvements may be made without departing from the scope of the present invention. For example, the rows and/or columns of the array of MCI keys may be slightly offset whilst retaining their overall shape.

The MCI key indicia can be adapted to suit the particular requirements of the application being used (i.e. different languages, computer-programming languages etc.). MCI keys can be arranged in different ways and layouts to cater for a variety of desktop needs, compactness, notebooks, portability and programmability etc. See, for example, the alternative layouts shown in Figs 12a-d.

In particular, Fig. 12c shows an alternative keyboard in which the priority order of most frequently occurring digraphs for English is [**th er on an re** he **in ed** nd ha **at en es of or**] with

demotion of [he, nd, ha] due to close association and occurrence with higher priority digraphs and relative occurrence with most frequently occurring tri-graphs for English [the and tha ent ion tio for nde has nce edt tis oft sth men], and exclusion of [of, or] due to least priority, frequency occurrence and limited real-estate on device.

Furthermore, the Trans- and Trans+ buttons included in the keyboards of Figs. 2a and 2b have been removed due to diversification of keyboard driver to operate for, and on, all conventional keyboard devices and software applications.

The MCI keys, as opposed to the conventional QWERTY keys have been re-sized in order to retain regular QWERTY look-and-feel of keys, and also optimise size accessibility and hit-ratio of the MDP-keys.

The Multi-Del and Multi-BSPC keys have been relocated to above the substantially central array of MCI-keys in order to streamline access and permit easy dissection of the keyboard (in a similar way to that shown in Fig. 2b)

Scientific calculator operators 1/x (reciprical of x) and x^y (x to power of y) are included to the right of the Function keys, while retaining internet TLDs [.biz and .pro] lost from exclusion of [of, or] MCI-keys.

The DUAL and DUAL SHIFT keys have been relocated beside the SPACE bars in order to inert any effect when accidentally hit relative to SPACE bar hits.

The MCI-keys have been re-arranged relative to home keys [F and J] where closest proximity is dependent on priority order of most frequent occurrence of digraph.